This application is submitted in the name of inventor Chintan Desai, a citizen of India, assignor to LSI Logic Corporation, a Delaware Corporation.

5

<u>S P E C I F I C A T I O N</u>

<u>TITLE OF INVENTION</u>

RECEIVE DESERIALIZER CIRCUIT FOR FRAMING PARALLEL DATA

10

<u>BACKGROUND OF THE INVENTION</u>

<u>Field of the Invention</u>

The present invention is generally directed to transceiver applications

15   involving transmitting and receiving parallel data. More specifically, the present

invention is directed to framing parallel data generated by a receive deserializer.

<u>Background</u>

In a typical transceiver application a transmitter on one chip multiplexes

20   parallel data, an eight bit parallel word for example, at one speed and transmits

this data as serial data at a higher speed. A receiver on another chip demultiplexes

the serial data and regenerates the eight bit parallel data word. However, the

receiver chip has no information as to which bit of the eight bit parallel word is bit

zero, the least significant bit (LSB), or which bit is bit seven, the most significant

25   bit (MSB). Thus, information regarding the boundary of the eight bit parallel

word has been lost in its transmission. The result is incorrectly framed parallel data words at the receiver which contain some bits belonging to the previous eight bit word or which contain some bits belonging to the next eight bit word.

5   This framing process is demonstrated generally in **FIG. 1**, which depicts a block diagram of a typical receiver deserializer along with a timing diagram showing how the received serial data is framed into parallel words based on a word clock. A sampling flip flop **100** receives serial data and retimes it with the rising edge of a recovered clock. The recovered clock is the source of eight phases

10   generated by a clock generator **104**. The serial data is sampled by the eight phases to generate eight bits. The eight bits are finally re-timed on one of the phases, channel word clock, to form a parallel data word. With bits B_7-B_0 making up the original parallel word which was serially transmitted, the timing diagram of **FIG. 1** illustrates how in a typical deserializer, the received data can be incorrectly

15   framed as a parallel data word.

Prior methods for solving this problem include the use of additional storage elements to store the last received eight bit word in order to create a new sixteen bit word from the last word and the current word. A block diagram of a

20   deserializer which implements this prior method is shown in **FIG. 2**. An array of eight storage flip flops **200** is used to store a previously received eight bit word which is then combined with a currently received eight bit word. The new sixteen

2

bit word contains a predefined data reference pattern which is transmitted upon initialization of the circuit. A comparator **202** searches through the new sixteen bit word for the predefined reference pattern and identifies the pattern location to a sixteen-to-eight multiplexer **204**. The sixteen-to-eight multiplexer **204** then selects these bits as the correct eight bits to be framed with word clock and output as received data.

5

The timing diagram of **FIG. 3** further illustrates this prior method for framing parallel data as implemented by the circuit of **FIG. 2**. Where bits B_7-B_0 represent an eight bit predefined reference pattern, the diagram indicates the combination of current and last data which forms a sixteen bit data word containing this predefined reference pattern. The bit locations of the predefined reference pattern within the sixteen bit data word are found through multiple comparisons made by the comparator **202** of **FIG. 2**. Once located by the comparator **202**, these bit locations are selected by the multiplexer **204** as containing the correct eight bits to be framed with word clock and output as received data.

10

15

This and other prior methods of framing parallel data can present significant costs in time and materials when implementing. For example, it is apparent from the timing diagram of **FIG. 3** that framing parallel data by the method presented in **FIG. 2**, introduces unwanted latency. There is a necessary

20

penalty of one word clock associated with the prior method of forming a sixteen bit data word to locate the predefined eight bit reference pattern.

Additionally, increased scrutiny of the block diagram circuit of **FIG. 2** indicates the complexity of the circuitry required to implement this prior method, as illustrated in **FIG. 4**. A circuit framework for the multiplexer **204** and comparator **202** blocks of **FIG. 2** is presented in **FIG. 4**. Though not intended as a complete representation of these circuit blocks, the depiction in **FIG. 4** shows the significant hardware required to implement the comparator **202** block of **FIG. 2**. Nine different sets of bit locations exist within the sixteen bit word where the predefined eight bit reference pattern might be encountered. For example, the predefined eight bit reference pattern could be located in bit locations 0-7, 1-8, 2-9, 3-10, 4-11, 5-12, 6-13, 7-14, or 8-15. It is therefore necessary to dedicate nine sets of eight comparators each to search these locations in order that the multiplexer **204** can select the correct location for framing the parallel data.

The disadvantages apparent in this and other prior methods of framing parallel data can therefore include requirements for additional data storage elements, complex comparator and multiplexer circuits, and unwanted latency inherent to these methods.

Accordingly, there exists a need for an efficient, simple and low latency method for framing the parallel data generated by a receive deserializer circuit.

SUMMARY OF THE INVENTION

A receive deserializer circuit which frames parallel data utilizes a skip-bit technique for aligning a predefined data reference pattern with a word clock. The receive deserializer circuit includes a sampling flip flop which receives serial data including a data reference pattern. The sampling flip flop samples and retimes the serial data to a recovered clock. A demultiplexer then deserializes the retimed serial data into a parallel data word which is timed to a word clock from a clock generator. A comparator makes comparisons of the parallel data word with a preset data reference pattern until a match results. A logic controller interprets whether the output of the comparator is a match and generates a shift pulse following each comparison which does not result in a match. The clock generator divides the recovered clock into eight phase clocks. One of the phase clocks is a word clock. Each time the clock generator receives a shift pulse from the logic controller, it disables all the phase clocks by one bit period. This results in a one bit shift in all the clocks and a one bit shift in the parallel data generated on word clock each time there is no match from the comparator. When a match occurs, no shift pulse is generated by the logic controller, and the predefined data reference pattern and subsequent data words are properly framed on the word clock.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block and timing diagram of a typical deserializer in accordance with the prior art illustrating a loss of word boundary information upon data transmission.

FIG. 2 is a block diagram of a deserializer circuit in accordance with the prior art illustrating a method for framing parallel data using additional storage elements.

FIG. 3 is a timing diagram illustrating the method for framing parallel data of the deserializer circuit of FIG. 2 in accordance with the prior art.

FIG. 4 is a diagram of the multiplexer and comparator portions of the deserializer circuit of FIG. 2 in accordance with the prior art, illustrating the complexity of implementing the comparator.

FIG. 5 is a block diagram of the deserializer circuit in accordance with a preferred embodiment of the present invention.

FIG. 6 is a diagram of the individual comparators within the comparator block of FIG. 5 in accordance with a preferred embodiment of the present invention.

FIG. 7 is a timing diagram illustrating the method for framing parallel data of the deserializer circuit of FIG. 5 in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Those of ordinary skill in the art will realize that the following description of the present invention is illustrative only and not in any way limiting. Other embodiments of the invention will readily suggest themselves to such skilled

5 persons after a perusal of this disclosure.

A preferred embodiment of the present invention is illustrated in the block diagram schematic of **FIG. 5**. In this illustration, control logic **500** is incorporated into a receive deserializer circuit similar to the circuit shown by the prior art of

10 **FIG. 2**. The control logic **500** initiates the skip-bit framing method of the present invention which eliminates the need for additional flip flop storage elements **200**, complex comparators **202**, and sixteen-to-eight multiplexers **204** as illustrated in the prior art of **FIG.'s 2 & 4**.

15 The receive deserializer circuit of **FIG. 5** comprises a sampling flip flop **502** which receives serial data from a transmitter circuit (not shown). The serial data includes a data reference pattern which is transmitted from the transmitter upon initialization of the circuit. The data reference pattern is typically, but not necessarily, an eight bit word. The sampling flip flop **502** retimes the serial data

20 to a recovered clock. The recovered clock is a clock which is local to the receiver circuit and which has been aligned on its negative edge with the transition edge of the incoming serial data by a clock recovery module (not shown).

A clock generator **506** divides down the recovered clock and generates eight phase clocks at a frequency eight times slower than the recovered clock. One of the phase clocks is word clock. The serial retimed data is sampled by each phase clock to generate eight bits which are retimed with word clock as parallel data by a demultipler **504**. In general, the parallel data word is compared by a comparator **508** which searches for the data reference pattern transmitted from the transmitter upon initialization of the circuit. The control logic **500** interprets whether the output of the comparator is a match and generates a shift pulse following each comparison which does not result in a match. When the clock generator **506** receives the shift pulse, it disables all the phase clocks by one bit period. This results is a one bit shift in the clocks with respect to the serial data and a one bit shift in the parallel data generated on word clock. Once the parallel data word contains all bits of the predefined data reference pattern, no shift pulse will be generated by the control logic **500**, and the parallel data word and all subsequent data words are properly framed on the word clock.

More specifically, a set of individual comparators **600** as shown in **FIG. 6** make up the comparator **508** of **FIG. 5**. The number of individual comparators **600** is typically eight, but in any event will be equal to the number of bits in the data reference pattern. The individual comparators **600** are typically operational amplifiers or logic gates such as exclusive OR gates. Each individual comparator

600 as shown in **FIG. 6**, has a first input tied to a preset bit representing a bit from the known data reference pattern. This preset bit is typically hardwired as an input to the individual comparator **600**, but could also be set upon initializing the receive deserializer circuit and stored in a storage element which then acts as the first

5 input to an individual comparator **600**. The second input to each individual comparator **600** is a corresponding bit from the parallel data word deserialized by the demultiplexer **504** of **FIG. 5**. The individual comparators **600** indicate a match when the preset bit and the corresponding bit from the parallel data word are the same. The control logic **500** effectively ANDs the individual comparator

10 **600** outputs together to determine if there has been a match between all the preset bits and all the bits in the parallel data word. If all the individual comparators **600** do not indicate a match at the same time, the control logic **500** generates a shift pulse to the clock generator **506** as indicated in **FIG. 5**. If all the individual comparators **600** indicate a match at the same time, the control logic **500** does not

15 generate a shift pulse, and the parallel data word is then known to contain the data reference pattern, which has thus been properly framed. The subsequent serial data being received on word clock is therefore also properly framed.

In addition to the demonstrated benefits of a reduced need for flip flop

20 storage elements, complex comparators, and sixteen-to-eight multiplexers, the timing diagram as shown in **FIG. 7** also illustrates the benefit of reduced latency that is gained by the framing method of the present invention. From the timing

diagram of **FIG. 7** it is apparent that if the word boundary and word clock is skewed by one bit period, the additional latency will only be one bit period. Furthermore, the maximum latency that can be generated by the framing method of the present invention as illustrated in **FIG.'s 5 & 7**, is four bit periods since the

5   word clock can be effectively shifted in either direction with respect to the sampled serial data and it can never be skewed more than four bit periods away from one of the word boundaries of an eight bit word. By contrast, the latency introduced by the prior art method as shown in the timing diagram of **FIG. 3** will always be one word clock cycle, regardless of the amount of initial skew in the

10   word clock and word boundary.

Alternative Embodiments

While embodiments and applications of this invention have been shown and described, it would be apparent to those skilled in the art that many more

15   modifications than mentioned above are possible without departing from the inventive concepts herein. The invention, therefore, is not to be restricted except in the spirit of the appended claims.